HN

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/607,592	06/30/2000	Dayman Pang	042390.P8716	1180

7590 01/19/2005

Donna Jo Coningsby
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP
7th Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025

EXAMINER	
BASOM, BLAINE T	
ART UNIT	**PAPER NUMBER**
2173	

DATE MAILED: 01/19/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/607,592	PANG, DAYMAN	
	Examin r	Art Unit	
	Blaine Basom	2173	

-- Th MAILING DATE of this communication appears on the cov r sh et with th correspondenc address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 October 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *22-41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *22-41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *30 June 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

The Examiner acknowledges the Applicant's cancellation of claims 1-21, and the addition of new claims 22-41. Regarding new claims 22-41, the Applicant submits that neither Chari (U.S. Patent No. 6,046,742), Takimoto (U.S. Patent No. 6,041,350), nor Kekic et al. (U.S. Patent No. 5,999,179) disclose all the elements of these new claims. In response, the Examiner respectfully presents the U.S. Patents of Dara-Abrams (U.S. Patent No. 6,456,892) and Shteyn (U.S. Patent No. 6,434,447), which as shown below, teach the elements recited in each of claims 22-41.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 22-30, 32-35, and 37-40 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,456,892, which is attributed to Dara-Abrams et al. (and hereafter referred to as "Dara-Abrams"). In general, Dara-Abrams describes a system by which a device, referred to as a "DDI controller," generates and displays a graphical user interface that is used to manage

and control a remotely-located device, which is referred to as a "DDI target" (for example, see column 4, line 35 – column 5, line 23).

Specifically regarding claims 22, 25-28, and 30, Dara-Abrams teaches developing and generating a graphical user interface (GUI), which is executed and displayed at the DDI controller (for example, see column 4, line 59 – column 5, line 8; column 6, line 60 – column 7, line 27; and column 11, line 66 – column 12, line 24). Dara-Abrams discloses that the DDI controller is not required to have any advanced-knowledge of the functionality or implementation of the DDI target device to display this GUI (see column 7, lines 27-48; and column 10, lines 48-67, for instance). Instead, the DDI controller delivers messages to the DDI target regarding the user's interaction with the GUI, wherein response, the DDI target interprets these messages and executes various functionality, possibly updating the state of the DDI target and also the display of the GUI (see column 12, line 38 – column 13, line 34; and column 19, line 53- column 20, line 44). Dara-Abrams is consequently considered to teach developing source code for a command-set unaware GUI with a graphical programming language, the GUI to be executed at a DDI controller, remotely from a DDI target. Moreover, Dara-Abrams discloses that the DDI target similarly comprises a user interface (for example, see column 3, lines 57-67), whereby it is understood that the logic used to implement this user interface is used to execute the DDI target functions in response to the above-described messages received from the DDI controller (for example, see column 7, lines 27-48; and column 12, line 38 – column 13, line 8). Dara-Abrams is consequently also considered to teach developing source code for this command-set aware user interface, the source code to be used to generate the user interface to execute at the remotely-located DDI target to configure a configuration parameter of the DDI

target according to a configuration parameter of the command set. Additionally, Dara-Abrams is considered to teach linking the source code for the command-set aware user interface in the source code for the command-set unaware GUI to provide a graphical component in the GUI associated with a configuration command. Dara-Abrams particularly discloses that an API is used to program the DDI controller to communicate with the DDI target, specifically to send messages to the target, which result in the target executing various commands, and to receive status updates from the target (for example, see "TABLE II" in column 22; and column 37, line 19 – column 38, line 59). Therefore, and with specific regard to claims 25-28, Dara-Abrams teaches providing a code hook, particularly an API, in the source code for the GUI to the DDI target. Such an API is understood to comprise a library of functions (for example, see "TABLE II" in column 22; and column 37, line 19 – column 38, line 59), which are referenced in the source code of the DDI controller's GUI. Each of these functions is considered a macro, like that of claims 27 and 28, which references an object-oriented class object in the source code of the GUI, the class object used to send messages to the DDI target, and consequently to invoke a routine of the source code of the DDI target. Also, since the target device executes various functionality in response to messages received from the DDI controller, the DDI target is considered to run "beneath" the DDI controller, and consequently, Dara-Abrams is considered to teach building the source code for the command-set unaware GUI with the linked source code for the command-set aware user interface to result in a device management application having the GUI with the command set aware user interface running beneath the GUI, remotely from the remote device, the GUI having the graphical component to provide access to the associated configuration command. It is understood that the source code for the GUI, along with APIs

linked to the source code for the DDI target device, are compiled to generate an executable binary device management application, like recited in claim 30. Dara-Abrams discloses that the DDI controller comprises a processor to process such source code (see column 9, lines 43-63). As is known in the art, the source code is necessarily translated into executable binary code so that the processor may understand the code. Consequently, Dara-Abrams is understood to teach a method, like that recited in claims 22, 25-28, and 30, which is for developing a device management application to manage a remote device.

Concerning claims 23 and 24, Dara-Abrams discloses that the DDI target, not the DDI controller, comprises the functionality to implement the various features of the target device, and to determine each of the states of the target device which occur in response to the executed features (see column 7, lines 27-48, for example). It is therefore understood that the user interface of the DDI target necessarily comprises code to execute every configuration command needed to generate every configuration state of the DDI target, like recited in claim 23. Specifically concerning claim 24, Dara-Abrams discloses that the DDI target maintains information regarding the configuration state of the DDI target device, and ascertains a new state of the device occurring in response to commands executed by the DDI target device (for example, see column 7, lines 27-48; column 12, line 60 – column 13, line34; and column 20, lines 23-45). It is therefore understood that the DDI target necessarily comprises one or more variables, data structures, or functions defining these configuration states of the DDI target, whereby such variables, data structures, or functions are configured according to commands executed at the DDI target.

With respect to claim 29, the source code for the DDI target interface is considered to be re-used, as it is linked it to the source code for the DDI controller's GUI, as is described above, and thus does not require the GUI's source code to comprise the same or similar code as the DDI target interface. Additionally, as Dara-Abrams discloses that this code for DDI target may be stored in ROM (see column 10, lines 18-27), it is understood that source code for the DDI target interface may comprise firmware. Dara-Abrams thus teaches that the source code for the DDI target interface may comprise firmware defining the user interface, whereby this firmware is re-used by linking it to the source code of the DDI controller's GUI defining a device management application.

As per claim 32, Dara-Abrams describes source code defining a user interface, considered a "console user interface" like the present application, which as described above, is generated and executed at a remote network device. Dara-Abrams discloses that the remote device, i.e. DDI target, maintains information regarding its configuration state, and ascertains a new state of the device occurring in response to commands executed by the DDI target device (for example, see column 7, lines 27-48; column 12, line 60 – column 13, line34; and column 20, lines 23-45). It is therefore understood that the DDI target necessarily comprises one or more variables, data structures, or functions, collectively considered a "configuration kernel" like that of the present application, which define the configuration state of the DDI target, and which are configured according to commands executed by the DDI target. Such commands may be performed in response to user interaction with the console user interface of the DDI target, or in response to user interaction with a GUI of a device management application executing at a DDI controller (for example, see column 14, lines 28-41; and column 20, lines 14-44). Regarding this

GUI, Dara-Abrams teaches that such a GUI is defined by source code comprising a hook, namely an API, which references the source code of the console user interface of the DDI target in order to provide a graphical component in the GUI to operate a function of the remote device and access a configuration command from the GUI, as is described above. As further described above, the user interface of the DDI target is considered to run "beneath" the GUI of the DDI controller, since the target device executes various functionality in response to messages received from the DDI controller GUI. Dara-Abrams is consequently considered to teach: receiving source code defining a console user interface (CUI) to generate a CUI to execute at a network device, the CUI to configure a configuration kernel (CK) of the network device according to a configuration command; receiving source code defining a GUI to generate a device management application, the GUI to be executed at a management point remote from the network device, the source code for the GUI to include a hook to the source code for the CUI to provide a graphical component in the GUI to operate a function of the GUI to access the configuration command from the GUI; and building the source code for the GUI with the hook to the source code for the GUI to create the device management application having the CUI running under the GUI, remotely from the remote device, the GUI having the graphical component to provide access to the associated configuration command. Accordingly, the development of such source code is understood to necessitate an article of manufacture comprising a computer-readable medium having instructions to cause a computer to perform operations like recited in claim 32.

As per claims 33 and 34, Dara-Abrams teaches providing a code hook, particularly an API, in the source code for the GUI to the source code for the CUI of the DDI target, as is

described above. Such an API is understood to comprise a library of functions, each associated with a particular class, and each relating to a message sent between the GUI and the DDI target (for example, see "TABLE II" in column 22; and column 37, line 19 – column 38, line 59). Dara-Abrams thus teaches providing source code defining the CUI as a library to the source code defining the GUI, wherein the source code for the GUI includes a hook to the source code for the CUI by including a reference to a class defined in this library, particularly, a function of a class defined in the library. Each of these functions is considered a macro, like that recited in claim 33, which references the source code of the CUI.

With respect to claim 35, the source code for the DDI target CUI is considered to be re-used, as it is linked it to the source code for the DDI controller's GUI, and thus does not require the GUI's source code to comprise the same or similar code as the DDI target CUI, as described above. Additionally, as Dara-Abrams discloses that the code for DDI target may be stored in ROM (see column 10, lines 18-27), it is understood that source code for the DDI target CUI may comprise firmware. Dara-Abrams thus teaches that the source code for the DDI target CUI may comprise firmware defining the user interface, whereby this firmware is re-used by linking it to the source code of the DDI controller's GUI defining a device management application.

Concerning claim 37, the DDI controller of Dara-Abrams comprises a GUI having a graphical element associated with a configuration command, whereby the graphical element is responsive to user input to operate the configuration command, and whereby the source code of the GUI references a code library, namely an API, which is associated with source code defining a CK and a CUI of a remote device, as is described above. It is understood that the CK may comprise a configuration parameter corresponding to a resource of the remote device, the

parameter indicating for example, a current state of the resource (for example, see column 12, line 60 – column 13, line 34, and column 20, lines 14-44). As further described above, the CUI of the DDI target interfaces this configuration kernel with the GUI of the DDI controller; the code library, i.e. API, is linked with the GUI and are understood to be necessarily compiled with the GUI to create a device management application having the GUI with the CK and CUI running under the GUI, remotely from the DDI target. The DDI controller of Dara-Abrams is additionally understood to comprise a communications interface coupled with the DDI target to communicate a configuration update for the DDI target from the device management application (for example, see column 19, lines 53-67). Moreover, the DDI controller of Dara-Abrams is understood to comprise a processor coupled with memory to implement the device management application and coupled with the communications interface to provide a user-selected configuration command to the interface (see, for example, column 9, lines 42-63; and column 19, lines 53-67). Consequently, the DDI controller of Dara-Abrams is considered an apparatus, like that of claim 37, which is for managing a remote device.

Referring to claims 38 and 39, Dara-Abrams particularly discloses that an API is used within the source code of the DDI controller to communicate with the DDI target, specifically to send messages to the target, which result in the target executing various commands, and to receive status updates from the target, as is described above. As further described above, this API is considered a library comprising a plurality of functions for communicating with the remote DDI target. Each of these functions is considered a macro, like recited in claim 38, and a hook to a subroutine, like recited in claim 39.

With respect to claim 40, Dara-Abrams discloses that the code for DDI target may be stored in ROM (see column 10, lines 18-27). It is therefore understood that source code for the DDI target CUI and CK may comprise firmware, developed for execution at the DDI target. The API of Dara-Abrams, which as described above communicates with the CUI and CK, is thus considered to include firmware code defining the CK and CUI, the firmware code developed for execution on the DDI target.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 31, 36, and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over the U.S. Patent of Dara-Abrams, which is described above, and also over U.S. Patent No. 6,434,447, which is attributed to Shteyn. As described above, Dara-Abrams teaches a method like that recited in claim 22, and an article of manufacture like that of claim 32, whereby the source code for a GUI executing at a controller device is linked to the source code for an interface executing at a remote device, so that the user of the controller device may implement the GUI to configure a configuration parameter of the remote device. Dara-Abrams, however, does not explicitly disclose updating configuration commands at the remote device, like recited in claim 41. In other words, Dara-Abrams does not teach: identifying an updated configuration command in the

configuration command set at the remote device; updating the source code for a command-set aware using interface at the remote device to reflect the updated configuration command in the configuration command set; and rebuilding the source code for the GUI at the controller device with the linked updated source code for the command-set aware user interface to result in an updated device management application having a graphical component to provide access to the updated configuration command, as is recited in each of claims 31 and 36.

Like Dara-Abrams, Shteyn describes a GUI executing at a controller device, with which a user interacts in order to manage and control a remote device located over a network (see column 2, lines 1-39 of Shteyn, for example). Regarding the claimed invention, Shteyn particularly teaches introducing new features and functionalities to the remote device, and accordingly, modifying the source code of the remote device to accommodate such new features and functionality (for example, see column 7, lines 6-18). Shteyn similarly teaches updating the GUI of the controller device, and particularly adding new user interface elements, understandably to access the new features and functionalities of the remote device (again, see column 7, lines 6-18).

Therefore, it would have been obvious to one of ordinary skill in the art, having the teachings of Dara-Abrams and Shteyn before him at the time the invention was made, to modify the system taught by Dara-Abrams, such that the source code of the remote device and the GUI of the controller device may be updated to add new features and functionality, as is taught by Shteyn. In other words, it would have been obvious to: add or update a configuration command in the configuration command set of the remote device of Dara-Abrams; to update the source code for the command-set aware user interface of the remote device to reflect the updated

configuration command in the configuration command set; and to rebuild the source code for the command-set unaware GUI of the controller device with the linked updated source code for the command-set aware user interface to result in an updated device management application having a graphical component to provide access to the updated configuration command. It would have been advantageous to one of ordinary skill to utilize this combination because such an ability to be updated facilitates "development and market penetration" of the remote and controller devices, as is taught by Shteyn (see column 7, lines 6-18).

Conclusion

The prior art made of record on form PTO-892 and not relied upon is considered pertinent to the Applicant's disclosure. The Applicant is required under 37 C.F.R. §1.111(C) to consider these references fully when responding to this action. The Karino U.S. Patent cited therein describes a network management system in which source code for a GUI executing at a management computer is linked to source code for a GUI executing at a remote network device, such that the GUI of the remote device runs beneath the GUI of the management computer.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Blaine Basom whose telephone number is (571) 272-4044. The examiner can normally be reached on Monday through Friday, from 8:30 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (571) 272-4048. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

btb

JOHN CABECA
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER